

Date of Deposit: October 28, 2003

Attorney Docket No. 14324US02

SOFTWARE SELF-REPAIR TOOLKIT FOR ELECTRONIC DEVICES

RELATED APPLICATIONS

[0001] This application makes reference to, claims priority to, and claims the benefit of United States Provisional Patent Application 60/428,071, entitled "Software Self-Repair Toolkit For Electronic Devices" (Attorney Docket No. 14324US01-101USMD20), filed November 21, 2002, the complete subject matter of which is hereby incorporated herein by reference, in its entirety.

[0002] In addition, this application makes reference to United States Provisional Patent Application Serial No. 60/249,606 (Attorney Docket No. 13776US01), filed November 17, 2000, PCT Application Serial No. PCT/US01/44034 (Attorney Docket No. 13776WO01), entitled "System And Method For Updating And Distributing Information", filed November 17, 2001, United States Patent Application Serial No. 10/311,462 (Attorney Docket No. 13776US02), entitled "System And Method For Updating And Distributing Information", filed 5/13/2003, and United States Provisional Patent Application Serial No. 60/410,071, entitled "Processing System With Component Architecture Platform Support" (Attorney Docket No. 14119US01-101USMD08), filed September 12, 2002, the complete subject matter of each of which is hereby incorporated herein by reference in its entirety.

FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

[0003] [Not Applicable]

[MICROFICHE/COPYRIGHT REFERENCE]

[0004] [Not Applicable]

BACKGROUND OF THE INVENTION

[0005] Electronic devices, such as mobile phones and personal digital assistants (PDA's), often contain firmware and application software that are either provided by the manufacturers of the electronic devices, by telecommunication carriers, or by third parties.

[0006] An update of firmware or firmware components typically involves procedures to be performed in a careful and error free manner. If errors occur, the device may be left in a partially or fully non-functional state. It is desirable to perform such updates using a fault tolerant mechanism. Fault tolerant code, however, is generally complex, and normally difficult to successfully implement.

[0007] Attempts to upgrade firmware and/or software in electronic devices, such as mobile phones, are often hampered by limited user interaction capabilities and slow communication speeds on these devices. In addition, when a user determines that there is an error with an electronic device, such as a mobile phones, the user often has limited recourse to having it corrected, and is often left with taking it back to a service center of the manufacturer to have the firmware/software modified or replaced.

[0008] Further limitations and disadvantages of conventional and traditional approaches will become apparent to one of skill in the art, through comparison of such systems with the present invention as set forth in the remainder of the present application with reference to the drawings.

BRIEF SUMMARY OF THE INVENTION

[0009] Aspects of the present invention may be seen in an updatable electronic device comprising a memory having at least one of firmware and software, a first firmware component for determining the occurrence of at least one of an error and an exception during the execution of the at least one of firmware and software, and a second firmware component for gathering at least one parameter related to the occurrence of the at least one of an error and an exception. Such an embodiment may also comprise interface circuitry for receiving update information via a communication network, the update information based upon the at least one parameter, and a third firmware component for applying the update information to at least a portion of the at least one of firmware and software. In an embodiment of the present invention, the at least one parameter may comprise at least one of a type of error, an error code, an error message, a type of exception, an identifier of a component, an identifier of a module, stack trace information, a computed hash value, and a device characteristic. The device characteristic may comprise at least one of a manufacture identifier, a model identifier, a firmware version number, a hardware version number, a battery level, a communication statistic, and a serial number. The device may be a mobile handset, and the communication network may comprise a wireless network. An embodiment of the present invention may comprise a fourth firmware component for generating an identifier for identifying the update information used to correct the at least one of an error and an exception.

[0010] In an embodiment of the present invention, the device may transmit the at least one parameter to at least one server via the communication network, the at least one server identifying the update information used to correct the at least one of an error and an exception. In various embodiments of the present invention, the update information may comprise information for retrieving at least one update package via the communication network, and a set of instructions for converting the at least a portion of the at least one of firmware and software from a first version to a second version. At least one of the first firmware component, the second firmware component, and the third firmware component may initiate at least one of a restart and a reboot of the device, and the device may store information related to the at least one of an error and an exception for use after at least one of

a power-up, a restart and a reboot of the device. The device may update the at least a portion of the at least one of firmware and software following at least one of a power-up, a restart and a reboot of the device. The at least one of firmware and software may comprise a java virtual machine.

[0011] Additional aspects of the present invention may be observed in a method of operating an updatable electronic device comprising a memory containing at least one of firmware and software. Such an embodiment may comprise determining whether at least one of an error and an exception has occurred, continuing regular processing if the at least one of an error and an exception has not occurred, and generating an identifier based upon at least one parameter if the at least one of an error and an exception has occurred. An embodiment of the present invention may also comprise receiving, via a communication network, update information based upon the identifier, and updating at least a portion of the at least one of firmware and software using the update information. The at least one parameter may comprise at least one of a type of error, an error code, an error message, a type of exception, an identifier of a component, an identifier of a module, stack trace information, a computed hash value, and a device characteristic. The device characteristic may comprise at least one of a manufacture identifier, a model identifier, a firmware version number, a hardware version number, a battery level, a communication statistic, and a serial number. The device may be a mobile handset, and the communication network may comprise a wireless network.

[0012] In an embodiment in accordance with the present invention, the device may transmit the at least one parameter to at least one server via the communication network, the at least one server identifying the update information used to correct the at least one of an error and an exception. The update information may comprise information for retrieving at least one update package via the communication network, and the update information may comprise a set of instructions for converting the at least a portion of the at least one of firmware and software from a first version to a second version. An embodiment of the present invention may also comprise initiating at least one of a restart and a reboot of the device. In addition, an embodiment of the present invention may comprise storing information related to the at least one of an error and an exception for use after at least one of a power-up, a restart and a reboot of the device. The device may update the at least a portion of the at least one of firmware and software following the at least one of a power-up, a restart and a reboot of the

device. Another embodiment of the present invention may comprise logging the occurrence of the at least one of an error and an exception, and communicating, via the communication network, the logged occurrence of the at least one of an error and an exception.

[0013] Yet another aspect of the present invention may be found in a method of operating an updatable electronic device. A embodiment in accordance with the present invention may comprise determining that at least one of a firmware and a software error has occurred, gathering at least one parameter related to the at least one of a firmware and a software error, and identifying update information for correcting the at least one of a firmware and a software error. Such an embodiment may also comprise receiving the update information via a communication network, and updating at least one of firmware and software using the update information. The communication network may comprise a wireless network, and the update information may comprise a set of instructions for converting the at least one of firmware and software from a first version to a second version.

[0014] Further aspects of the present invention may be observed in a network for updating at least one of firmware and software in at least one updatable electronic device. Such a network may comprise at least one server receiving at least one of device characteristics and an error report from the at least one updatable electronic device, the at least one server retrieving appropriate update information based on the at least one of device characteristics and an error report. The at least one server may communicate the update information to the at least one updatable electronic device using a wireless interface, and the at least one updatable electronic device may use the update information to update the at least one of firmware and software. The update information may comprise an update package having a set of instructions for converting the at least one of firmware and software from a first version to a second version, and the updatable electronic device may comprise a mobile handset. The device characteristics may comprise at least one of a manufacture identifier, a model identifier, a firmware version number, a hardware version number, a battery level, a communication statistic, and a serial number. The updatable electronic device may also comprise a display for displaying information regarding at least one of an error, an exception, and an event to a user. The displayed information may comprise a user prompt to solicit user approval for at least one of the communication of the update information and the update of the at least one of firmware and software.

[0015] These and other advantages, aspects and novel features of the present invention, as well as details of an illustrated embodiment thereof, will be more fully understood from the following description and drawings.

BRIEF DESCRIPTION OF SEVERAL VIEWS OF THE DRAWINGS

[0016] Fig. 1 is block diagram of an electronic device, comprising a software self-repair toolkit, a device characteristics module, a loaders module, a component architecture platform module, an optional security module, an optional application and an optional java virtual machine (JVM), in accordance with an embodiment of the present invention.

[0017] Fig. 2 is a flow chart depicting an exemplary method of operating a software self-repair toolkit, such as the software self-repair toolkit, in an electronic device, such as the electronic device of Fig. 1, in accordance with an embodiment of the present invention.

[0018] Fig. 3 is a flow chart depicting an exemplary method of operating a software self-repair toolkit of an electronic device, such as the software self-repair toolkit of the electronic device of Fig. 1, when a new feature is invoked for the first time, in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0019] The present invention relates generally to the process of updating software/firmware in electronic devices and, more specifically, to the use of a software self-repair toolkit that facilitates the download of update packages and the subsequent update of firmware/software employing an update package.

[0020] Fig. 1 is block diagram of an electronic device 109, comprising a software self-repair toolkit 123, a device characteristics module 129, a loaders module 137, a component architecture platform module 143, an optional security module 145, an optional application 131 and an optional java virtual machine (JVM) 141, in accordance with an embodiment of the present invention. An electronic device such as electronic device 109 of Fig. 1 may employ the software self-repair toolkit 123 to trap errors and/or exceptions encountered during the execution of its firmware/software, and to identify, download, and apply update packages to fix the firmware/software bugs that cause such errors or exceptions.

[0021] The software self-repair toolkit 123 of Fig. 1 comprises an error/exception handler module 135, an error identity computation module 127, an authentication unit 133, an update agent 125, and a manager 147. The software self-repair toolkit 123 may employ any of the loaders supported by the loaders module 137 to retrieve update packages from an external system such as, for example, the server 119 of the carrier network 107. The software self-repair toolkit 123 may also store information about downloaded update packages, such that the update agent 125 can retrieve them for subsequent use.

[0022] In an embodiment of the present invention, when the electronic device 109 encounters an error and/or exception during execution of elements of its firmware/software (e.g., the java virtual machine (JVM) environment 141, component architecture platform 143, application 131, etc.), the details of the error and/or exception may be saved, and subsequently used to help identify an appropriate firmware/software update, to correct the occurrence of the error and/or exception. An appropriate update package to correct the problem may then be retrieved from a server 119 to which the electronic device 109 is communicatively coupled. The manager 147 of the software self-repair toolkit 123 may coordinate retrieval of the update package based upon one or more parameters such as, for example, parameters used to

identify the type of error, parameters used to identify a component or module involved, device characteristics used to identify the device and the device configuration, etc. The manager 147 may retrieve, using the above parameters, one or more update package(s) identified as likely to correct the firmware/software bug causing the error and/or exception. The manager 147 of the software self-repair toolkit 123 may then store status information, set flags, save addresses for the locations where the update package(s) are stored, etc., and may restart/reboot the electronic device 109. The update agent 125 of the software self-repair toolkit 123 may then be employed to apply the update package(s) to the firmware/software of the electronic device 109 (i.e., application 131, JVM environment 141, etc.) The update package(s) may comprise, for example, a set of instructions for converting the firmware/software of the electronic device 109 from a first version to a second version. A description of such an approach may be found in United States Provisional Patent Application Serial No. 60/249,606 (Attorney Docket No. 13776US01), filed November 17, 2000, PCT Application Serial No. PCT/US01/44034 (Attorney Docket No. 13776WO01), entitled "System And Method For Updating And Distributing Information", filed November 17, 2001, and United States Patent Application Serial No. 10/311,462 (Attorney Docket No. 13776US02), entitled "System And Method For Updating And Distributing Information", filed 5/13/2003, the complete subject matter of each of which is hereby incorporated herein by reference, in its entirety.

[0023] The following description addresses three possible scenarios, characterized primarily by the capabilities of the execution environment, i.e. the functionality of electronic device 109. In each of these scenarios, an embodiment of the present invention facilitates the detection and identification of the error, identification and retrieval of appropriate update package(s), and the application of the retrieved update package(s). In addition, an embodiment of the present invention may verify and/or authenticate the update package(s), before the update is performed by the update agent 125.

[0024] In one embodiment of the present invention, the electronic device 109 may employ a single threaded environment that becomes partially inoperative when an error and/or exception is encountered. In such a situation, the electronic device 109 may be incapable of retrieving an update package without restarting/rebooting. In an embodiment of the present invention, an exception handler such as the exception handler 135 of Fig. 1 may save an

appropriate set of parameters, identifiers, and error details about the error and/or exception that was encountered. The exception handler 135 may select a different combination of parameters, identifiers, and error details based upon the error and/or exception that occurred.

[0025] An embodiment of the present invention may employ the error identity computation module 127 to map information about the error to a unique identifier such as, for example, a computed hash value, that may be used to retrieve an update package. The update package is selected for retrieval based upon the likelihood that it will fix the firmware/software bug that caused the error and/or exception to occur. Such an embodiment may also record the occurrence of the error and/or exception for subsequent discovery. After a restart/reboot, an embodiment of the present invention may detect the earlier occurrence of an error and/or exception, access the parameters, identifiers, and error details that were recorded, and retrieve appropriate update package(s). Such an embodiment may use an update agent, such as update agent 125 of Fig. 1, to update the errant firmware/software using the retrieved update package(s). The update agent 125 may then initiate a restart/reboot of the electronic device 109, in order to complete the correction of the firmware/software of the electronic device 109.

[0026] In another embodiment of the present invention, the electronic device 109 may employ a multi-threaded operating system. The electronic device 109 may be, for example, a cell phone equipped with the Symbian operating system, as offered by Symbian, Ltd. In such an environment, the software self-repair toolkit 123 of the electronic device 109 may employ the error/exception handler 135 to gather information and/or symptoms of the error and/or exception. An embodiment of the present invention may employ the error identity computation module 127 to either generate an identifier based upon the gathered information and/or symptoms of the error and/or exception, to identify an appropriate update package. In a related embodiment, the software self-repair toolkit 123 may communicate the symptoms and device characteristics 129 to a remote server, such as server 119 in carrier network 107. Communication of the symptoms and device characteristics to the server 119 may allow the server 119 to determine an appropriate update package for the electronic device 109. In either of these embodiments, the software self-repair toolkit 123 may communicate the gathered information/symptoms or the identifier and associated info to the server 119 using a second, independent thread or process for such communication. The software self-repair

toolkit 123 may subsequently retrieve appropriate update package(s) employing one of the loaders available in the loaders module 137. One such loader in loaders module 137 may be, for example, a universal resource locator (URL) loader. Following receipt of the appropriate update package(s), the electronic device 109 may employ the authentication unit 133 to verify and/or authenticate the retrieved update package(s), and employ the update agent 125 to apply the received update package(s) to the associated firmware/software of electronic device 109. The software self-repair toolkit 123 may then initiate a restart/reboot of the electronic device 109 to complete the correction of the errant firmware/software.

[0027] In yet another embodiment of the present invention, the electronic device 109 may comprise a JVM environment 141 and the software self-repair toolkit 123 may detect error(s) and/or exception(s) encountered by the JVM environment 141 during the execution of programs. When an error occurs or an exception is encountered by the JVM environment 141, the software self-repair toolkit 123 may employ the error/exception handler 135 to gather information and/or symptoms of the error/exception. The software self-repair toolkit 123 may employ the error identity computation module 127 to generate an identifier based on such gathered information (to be used to retrieve appropriate update package(s)), or to communicate the symptoms and device characteristics to a server, such as the server 119 of carrier network 107. The server 119 may then identify one or more appropriate update package(s), retrieve the identified update package(s), and provide the appropriate update package(s) to the electronic device 109. The update agent 125 of the software self-repair toolkit 123 of electronic device 109 may then be employed to apply any update package(s) received.

[0028] In a related embodiment of the present invention, stack information retrieved by the printStackTrace() method of the JVM environment 141 may be saved by the error/exception handler 135, along with other information related to the error and/or exception. The error identity computation module 127 may then identify and retrieve a java archive (jar) file as an update package. The retrieved update package may be verified and/or authenticated by the authentication unit 133. The update agent 125 may then update the classpath of the JVM environment 141 to incorporate the retrieved jar file.

[0029] In a further embodiment of the present invention, a component architecture platform (CAP) module, such as CAP module 143 of Fig. 1, may maintain a catalog of components in

the electronic device 109, with details of each component. Component details may include, for example, the starting address, ending address (if appropriate), the size (such as number of blocks), version numbers (major, minor etc.), CRC values, and other related information, for each of the cataloged components in the CAP module 143. A description of one such component architecture platform is provided in United States Provisional Patent Application Serial No. 60/410,071, entitled “Processing System With Component Architecture Platform Support” (Attorney Docket No. 14119US01-101USMD08), filed September 12, 2002, the complete subject matter of which is hereby incorporated herein by reference in its entirety. When the update agent 125 of the software self-repair toolkit 123 is to apply a retrieved update package to one of the components of the CAP module 143, the update agent 125 may retrieve component details such as the starting address, CRC values, etc. from the catalog of components maintained by the CAP module 143.

[0030] In one embodiment of the present invention, the manager 147 of the software self-repair toolkit 123 may maintain a catalog of known errors or exceptions, and a list of update packages that may be used to fix the known error or exception. When the error/exception handler 135 encounters an error or exception, the error identity computation module 127 may compute a unique identifier (e.g., a computed hash value) for the error or exception based upon details of the error or exception (such as stack trace, error messages, error codes, etc.), and may employ the unique identifier to determine whether an update package containing a bug fix exists. The software self-repair toolkit 123 may retrieve the identified update package(s) employing one of the loaders provided by the loaders module 137. The update agent 125 may then be employed to apply the retrieved update package(s) to the firmware/software of electronic device 109. In one embodiment of the present invention, the update agent 125 may be invoked after a reboot of the electronic device 109. The determination of the existence of an update package based on the unique identifier computed by the error identity computation module 127 may be conducted by the manager 147 in a related embodiment of the present invention, and by the server 119 in yet another embodiment.

[0031] In an embodiment of the present invention, when a feature of the electronic device 109 is used for the first time, the manager 147 may determine whether a potential firmware/software bug exists. If potential firmware/software error(s) are identified, the

manager 147 may then determine whether an update package containing a fix for the identified firmware/software error(s) exists. If the manager 147 determines that an update package is available to correct the firmware/software bug, the manager 147 may automatically retrieve the identified update package from the server 119. The manager may verify and/or authenticate the update package, and apply the update package to the firmware/software of electronic device 109, using the update agent 125. Following correction of the firmware/software error, the electronic device 109 may invoke the feature for the user. Feedback to the user such as, for example, a progress bar, may be displayed during the retrieval of an update package, and during the application of the update package by the update agent 125.

[0032] In one embodiment of the present invention, the electronic device 109 may be a mobile handset that may save a small piece of information (such as, for example, a feature identifier) whenever any feature of an application (e.g., a firmware/software component, a device driver, or any application firmware/software) is invoked. For example, a unique feature or application identifier (e.g., a computed hash value) may be saved, along with additional information available at the time of invocation. The unique feature identifier and the additional information may be stored in memory such as, for example, FLASH memory, or other types of non-volatile storage, just prior to invocation of the feature. This saved information may be referred to as a “feature trace”. If the mobile handset then becomes inoperable due, for example, to an error or an exception encountered while executing any component of the firmware/software of the feature, the mobile handset may identify the feature that caused the error and/or exception, and take corrective actions. This may be particularly effective if the error and/or exception is flagged, and the saved information is processed during a subsequent restart/reboot. Thus, the fact that an error was encountered in a previous invocation may be displayed when a feature is invoked, along with a prompt to the user to either retrieve an update package that fixes the bug, or to ignore the message and invoke the application or feature despite the warning. In a related embodiment, the update of the application, component, etc. may be accompanied by the simultaneous deletion of the saved feature or application identifier and additional information, in order to avoid warnings to the user after the error-causing bug has been fixed by an update activity.

[0033] A feature identifier may be tracked to determine the feature that caused an error or exception while running an application and subsequent invocation of the feature may be preceded by the display of an appropriate warning and an opportunity to retrieve and apply an update package that fixes the bug. In a related embodiment, a combination of an application identifier and a feature identifier may be saved to track where in an application an error or exception occurred. Saving other types of identifiers and related data are also contemplated.

[0034] In an embodiment of the present invention, a device characteristics module, such as the device characteristics module 129 of Fig. 1, may support the retrieval of device characteristics that are static, as well as those that are dynamic in nature. Static device characteristics that may be stored in device characteristics module 129 may comprise attributes of a mobile handset such as, for example, a unique identifier (e.g., a computed hash value), manufacturer information, model information, firmware version number(s), hardware version numbers(s), and the like. Dynamic device characteristics that may be stored in a device characteristics module may comprise information such as battery level(s), and communications statistics (e.g., the number of characters transmitted or received, the number of communication errors detected, the ratio of dropped packets in a communication link, etc.)

[0035] Fig. 2 is a flow chart depicting an exemplary method of operating a software self-repair toolkit, such as the software self-repair toolkit 123, in an electronic device, such as the electronic device 109 of Fig. 1, in accordance with an embodiment of the present invention. In the method illustrated in Fig. 2, when an error or exception is encountered, the software self-repair toolkit in the electronic device retrieves an update package and applies the update package to fix the error or exception encountered. The operation of the electronic device begins when it is powered-up or restarted/rebooted (block 207). Next, the execution of a feature/application is started (block 209). This typically occurs based upon user input. Then, the occurrence of an error or exception is determined (block 211). If it is determined that an error or exception has not been encountered, the regular processing of the feature/application is executed (block 225), and then feature/application processing terminates (block 221).

[0036] If it is determined that an error or exception has been encountered (block 211), then an error identity computation module, such as error identity computation module 127 of Fig. 1, may generate or compute a unique identifier (block 213). The error identity computation module 127 may employ an error code, error message and/or stack trace information to

generate or compute the unique identifier (e.g., a computed hash value) that is subsequently employed to determine if an update package exists at a server, such as server 119. The unique identifier may be used to retrieve an update package containing a bug fix for the encountered error or exception. In another embodiment of the present invention, the error identity computation module 127 may gather an error code, error message and/or stack trace information and communicate that to a server. The server may use the received error code, error message and/or stack trace information for the identification and retrieval of an appropriate update package(s).

[0037] Next, an attempt may be made to determine if an update package exists at a server (block 215). For example, in one embodiment, a computed or generated unique identifier (e.g., a computed hash value) may be employed to determine if an update package exists at the server. The unique identifier may be generated by and communicated to the server by the electronic device. Other information, such as device characteristics, may also be communicated. In another embodiment, an error code, error message and/or stack trace information may be gathered and sent to the server, to be used for the identification and retrieval of an appropriate update package(s). If it is determined that an update package does not exist for the error or exception encountered (block 215), then an the error is logged (block 223), and regular processing may resume (block 207). In one embodiment of the present invention, the electronic device may communicate to the server a message containing the logged error, using an appropriate communication link. Examples of appropriate links include wireless application protocol (WAP) links, or http links, employing an appropriate format such as, for example, a SyncML message format. Such protocols may be used over wireless networks such as, for example, cellular or personal communications service (PCS) networks.

[0038] If it is determined that an update package does exist (block 215), then the update package(s) is/are retrieved and saved in the electronic device (e.g., in FLASH storage, in a file system such as the one provided by the Symbian operating system, etc.) after the authenticity of the package(s) have/has been verified. Later, the retrieved update package(s) may be used by an update agent, such as update agent 125 of Fig. 1, to update the feature/application software (block 219).

[0039] The update agent may also use information about the feature/application maintained by a CAP module, such as CAP module 143. In one embodiment of the present invention, the server may communicate the need to reboot the electronic device once the update process is complete. In such an embodiment, the update agent may be capable of accessing such information and acting on it. At the end of the update process, regular processing of the feature/application resumes (block 207).

[0040] Fig. 3 is a flow chart depicting an exemplary method of operating a software self-repair toolkit of an electronic device, such as the software self-repair toolkit 123 of the electronic device 109 of Fig. 1, when a new feature is invoked for the first time, in accordance with an embodiment of the present invention. In the illustration of Fig. 3, the electronic device attempts to determine whether an error or exception is likely to be encountered and/or a bug fix or update is to be applied. The operation of the electronic device begins when it is powered-up or restarted/rebooted (block 307). Next, the execution of a feature/application is started (block 309). This is normally initiated by user input. Then, the electronic device determines whether a previously unused (possibly new) feature/application is being invoked (block 311). If it is determined that the feature/application being invoked has been used before, the processing for that feature/application is executed (block 325), and then the feature/application processing terminates (block 321).

[0041] If it is determined that a feature/application is being invoked for the first time (block 311), then an error catalog may be accessed to identify firmware/software bugs that have been identified for that feature/application, and for which associated update packages may be available (block 313). In one embodiment of the present invention, the error catalog may be maintained by a server, such as the server 119 of Fig. 1. In such an embodiment, a manager, such as the manager 147 of Fig. 1, may communicate with the server to determine if a known error or exception is likely to be encountered when the newly accessed feature is exercised. In another embodiment, the error catalog may be maintained by the manager, and may be accessed to determine whether an error or exception is likely to be encountered.

[0042] Next, an attempt is made to determine whether an update package exists (block 315). In one embodiment of the present invention, a unique identifier (e.g., one that is retrieved from the known error catalog maintained in the manager or by the server) may be employed

to determine whether an update package exists at the server. In such an embodiment, the unique identifier may be communicated to the server by the electronic device, along with other information (e.g., device characteristics, such as the device characteristics 129 of Fig. 1).

[0043] If it is determined that an update package does not exist for an error or exception likely to be encountered in the newly invoked feature/application (block 315), then an error is logged, recording the fact that an update package is not available (block 323). Regular processing of the new feature/application then resumes (block 325). In one embodiment of the present invention, the electronic device may communicate to the server a message containing the logged error, using an appropriate communication link. Examples of appropriate links include wireless application protocol (WAP) links, or http links, employing an appropriate format such as, for example, a SyncML message format. Such protocols may be used over wireless networks such as, for example, cellular or personal communications service (PCS) networks.

[0044] If it is determined that an update package does exist (block 315), then the update package(s) is/are retrieved and saved in the electronic device (e.g., in FLASH storage, in a file system such as the one provided by the Symbian operating system, etc.) after the authenticity of the package(s) has/have been verified. Later, the retrieved update package(s) may be used by an update agent, such as update agent 125 of Fig. 1, to update the application software (block 319).

[0045] The update agent may also use information about the feature/application maintained by a CAP module, such as CAP module 143. In one embodiment of the present invention, the server may communicate the need to reboot the electronic device once the update process is complete. In such an embodiment, the update agent may be capable of accessing such information and acting on it. At the end of the update process, regular processing of the feature/application resumes (block 307).

[0046] In one embodiment of the present invention, the software self-repair toolkit 123 of Fig. 1 may be adaptive in nature, and may keep track of all errors and exceptions encountered, the update packages that have been retrieved and/or applied, the status of the features/applications that have been updated, the information related to components that have been updated, etc.

[0047] While the invention has been described with reference to certain embodiments, it will be understood by those skilled in the art that various changes may be made and equivalents may be substituted without departing from the scope of the invention. In addition, many modifications may be made to adapt a particular situation or material to the teachings of the invention without departing from its scope. Therefore, it is intended that the invention not be limited to the particular embodiment disclosed, but that the invention will include all embodiments falling within the scope of the appended claims.